Demographics at Higher Risk from Future Pandemics

Group 8

Utkarsh Singh
Fenil Savani

857-313-0037
854-844-9760

singh.utka@northeastern.edu
savani.f@northeastern.edu

Percentage of Effort Contributed by Student 1: _____50%___________

Percentage of Effort Contributed by Student 2: _____50%______

Signature of Student 1: Utkarsh Singh

Signature of Student 2: Fenil Savani

Submission Date: 21 April 2023

Contents

Problem Setting

The current world demographics is changing at an alarming pace over the past 10 years. Directly impacted by ever changing climate and rising costs across the globe – this inadvertently also has led to severe complications with the new diseases amongst the people. The outbreak of COVID-19 (coronavirus SARS-CoV-2) posed major problems to the people across the globe. Complications arising due to geographical location, population density, age group of people, previous health complications (co morbidity), climate, etc. made it difficult for the various agencies across the world to track and prevent the spread of the virus.

Problem Definition

The project analysis aims to provide the best classification models and predictors so as to correctly predict the future demographics that are at the highest risk of being impacted from a future virus strain similar to COVID (Airborne transmission). The main intention is to identify the counties in US that are at a higher risk from airborne transmission of virus and help them prepare beforehand for such situations.

Data Source

The data source is taken from Kaggle repository (https://www.kaggle.com/datasets/johnjdavisiv/us-counties-covid19-weather-sociohealth-data).

Data Description

The dataset comprises of 790330 instances. There are a total of 227 entries and 1 target attribute – 'Cases', which indicates the number of cases recorded on a particular date in a particular county.

The Dataset is mostly consisting of records that are numerical – but also has categorical data such as stay_at_home_effective, county and state. Out of the various numerical attributes our main focus will lie to use specific columns out of the 227 that would provide us valuable information for the demographics such as total_population, area_sqmi, percent_smokers, etc.

Data Mining Tasks

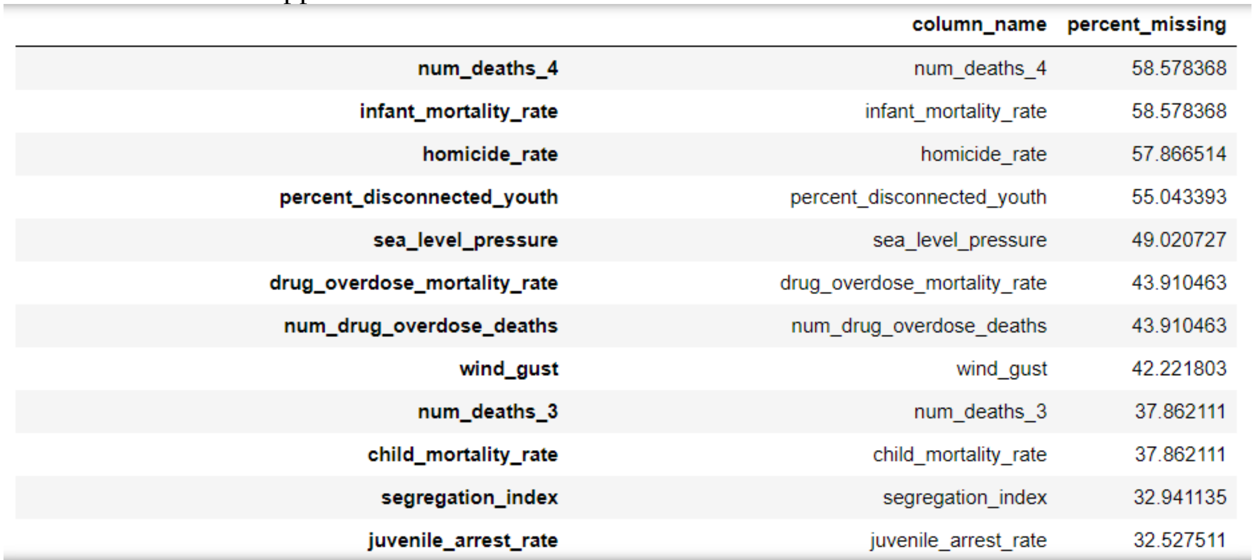
a. Data Understanding

The initial dataset comprised of 790331 records/observations and 227 Attributes, and one target variable **Case.** On analysis it was found out that the in the entire dataset there were only 3 categorical attributes – Date, stay at home announced and stay at home effective. Stay at home announced & stay at home effective which is a **Categorical data** are binary in nature i.e., with only 2 different types of observations i.e., **YES & NO**

In order to make the data uniform as possible (numerical observations for maximum attributes) – **Labelling and Coding** was performed on column 'stay at home announced' and 'stay at home effective' i.e., YES and NO were converted into 1 and 0 respectively.

b. Data Pre-processing

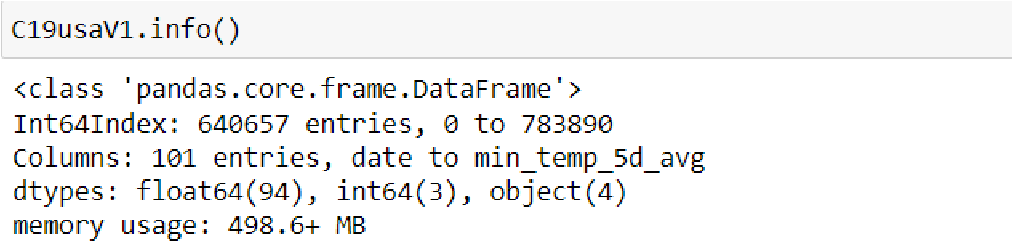
Since the dataset consisted of 790331 records and 227 columns – it was necessary to clean the data as all 227 columns would not serve any purpose. Initial analysis into the dataset focused on the columns with the most missing values. It was found out that 20 columns have more than 20% null values (5 Columns had 50% missing values). These 20 columns when heavily affecting into the factor of rows with at least one missing observation; 756024 rows were having at least one missing value i.e., **95% of the observations were missing one value**. The 20 attributes were dropped.



	column_name	percent_missing
num_deaths_4	num_deaths_4	58.578368
infant_mortality_rate	infant_mortality_rate	58.578368
homicide_rate	homicide_rate	57.866514
percent_disconnected_youth	percent_disconnected_youth	55.043393
sea_level_pressure	sea_level_pressure	49.020727
drug_overdose_mortality_rate	drug_overdose_mortality_rate	43.910463
num_drug_overdose_deaths	num_drug_overdose_deaths	43.910463
wind_gust	wind_gust	42.221803
num_deaths_3	num_deaths_3	37.862111
child_mortality_rate	child_mortality_rate	37.862111
segregation_index	segregation_index	32.941135
juvenile_arrest_rate	juvenile_arrest_rate	32.527511

Figure 1. The Missing values from the initial dataset in Descending order

After dropping the 20 attributes – the **missing value observations dropped by 60%.**
Post dropping 20 attributes based on missing values – we proceed ahead to segregate attributes not required for the particular problem definition and drop them based on domain knowledge. In total we **dropped additional 97 Attributes** (attributes such as weather, number of college pass outs, etc.) After dropping all the attributes not required – we look at the number of observations with at least one missing value. On calculation, the number of **rows with at least one missing value is 149674 – 18.9% of the total observations**. The missing observations well below the 20% margin – the entire missing observations are dropped.
Finally, we are left with **101 attributes and 640657 observations.**



```
C19usaV1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 640657 entries, 0 to 783890
Columns: 101 entries, date to min_temp_5d_avg
dtypes: float64(94), int64(3), object(4)
memory usage: 498.6+ MB
```

Figure 2. The Info on the data post the second round of preprocessing.

c. Correlation analysis using Pearson's Correlation

From the heatmap below, using Pearson's correlation matrix we can see that a few attributes have very high correlation amongst them – some even close to 0.95

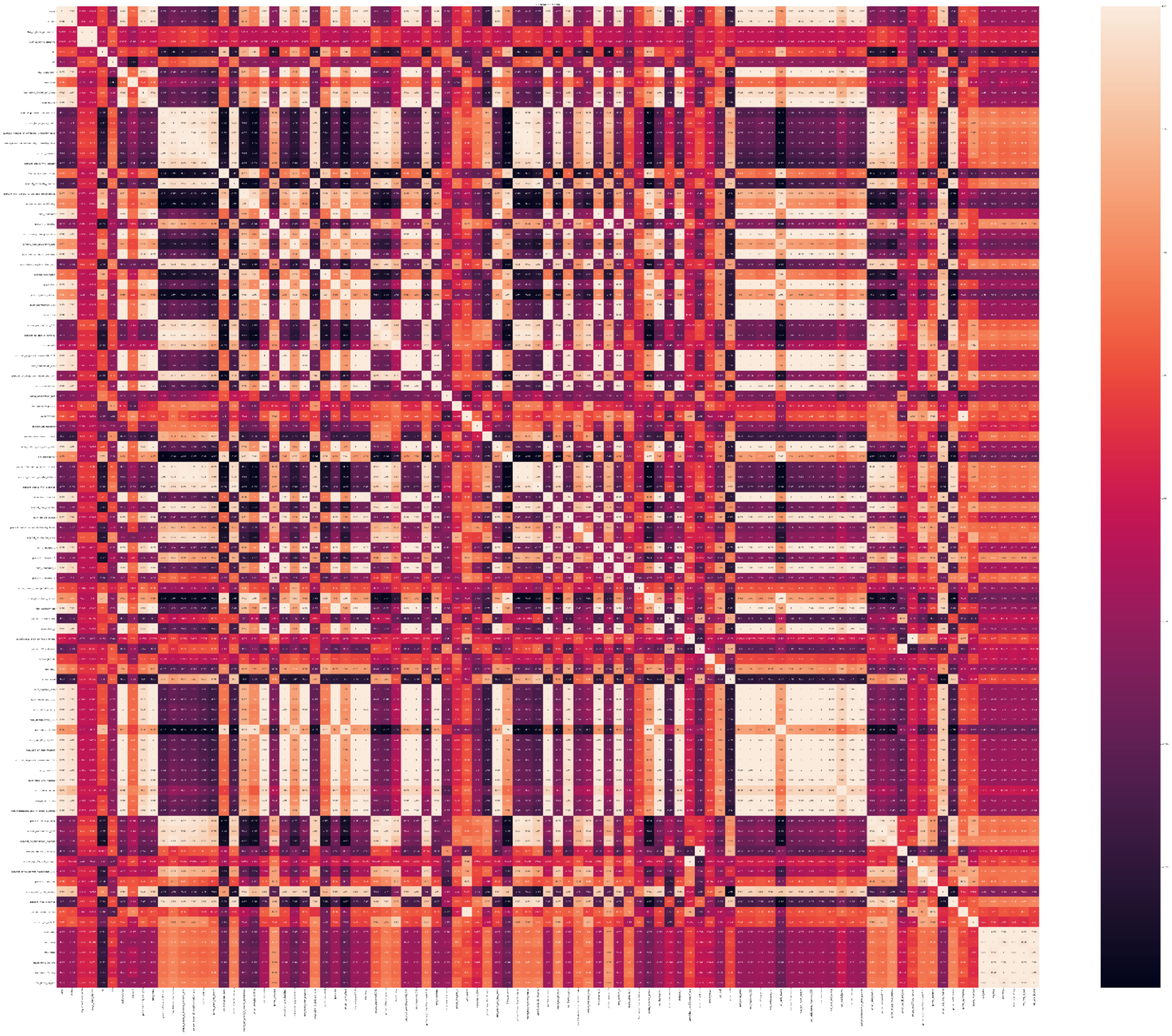


Figure 3. The Correlation Matrix for the 101 columns

Visualizations

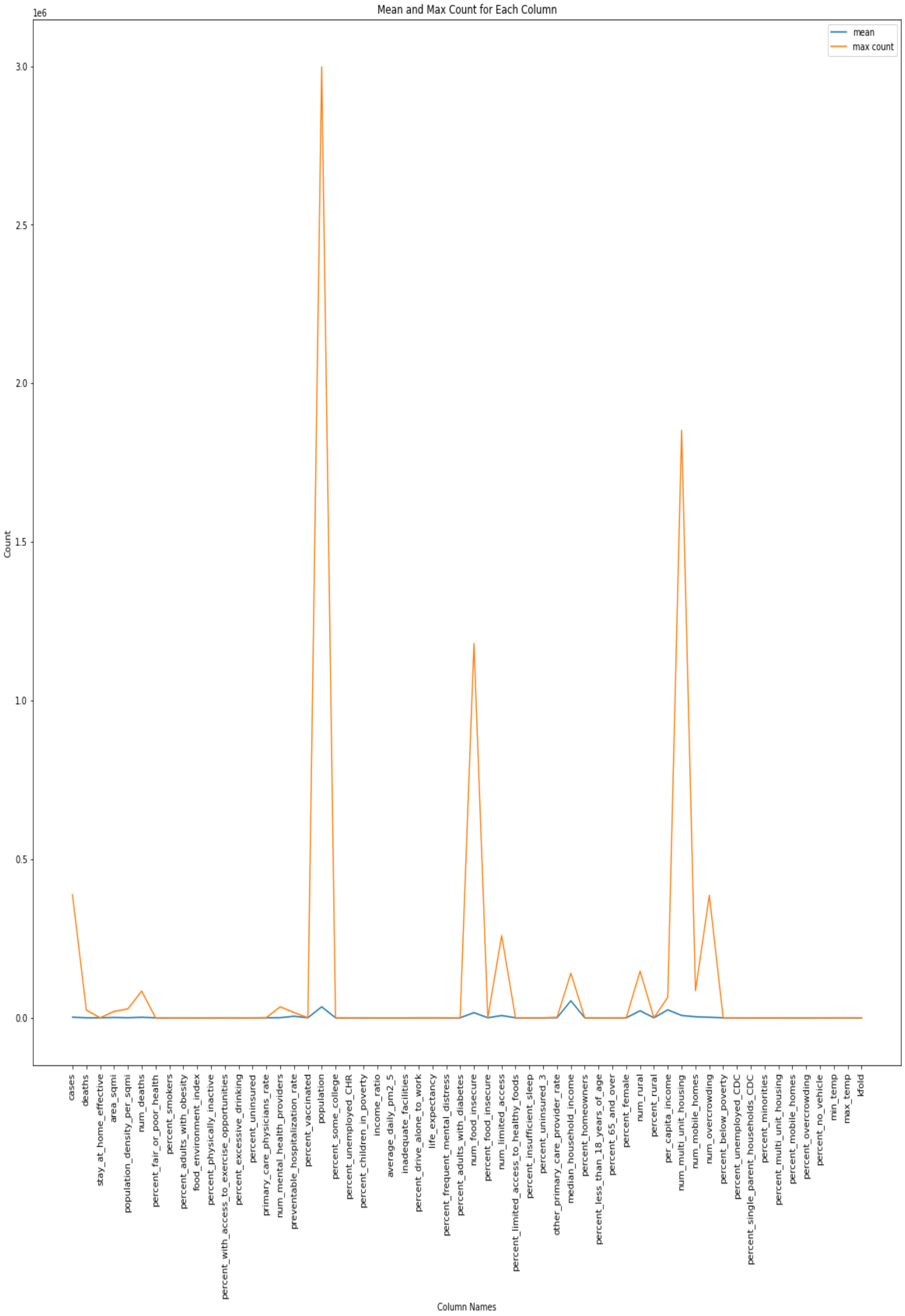


Figure 4. The above chart describes the difference between the Mean and the Highest count in Each Attribute

Over the course of the final columns selected – we could easily see the difference between the mean and the max value in multiple columns. This does confirm the presence of large outliers in the dataset. This further in the project impacts our process of handling the data.

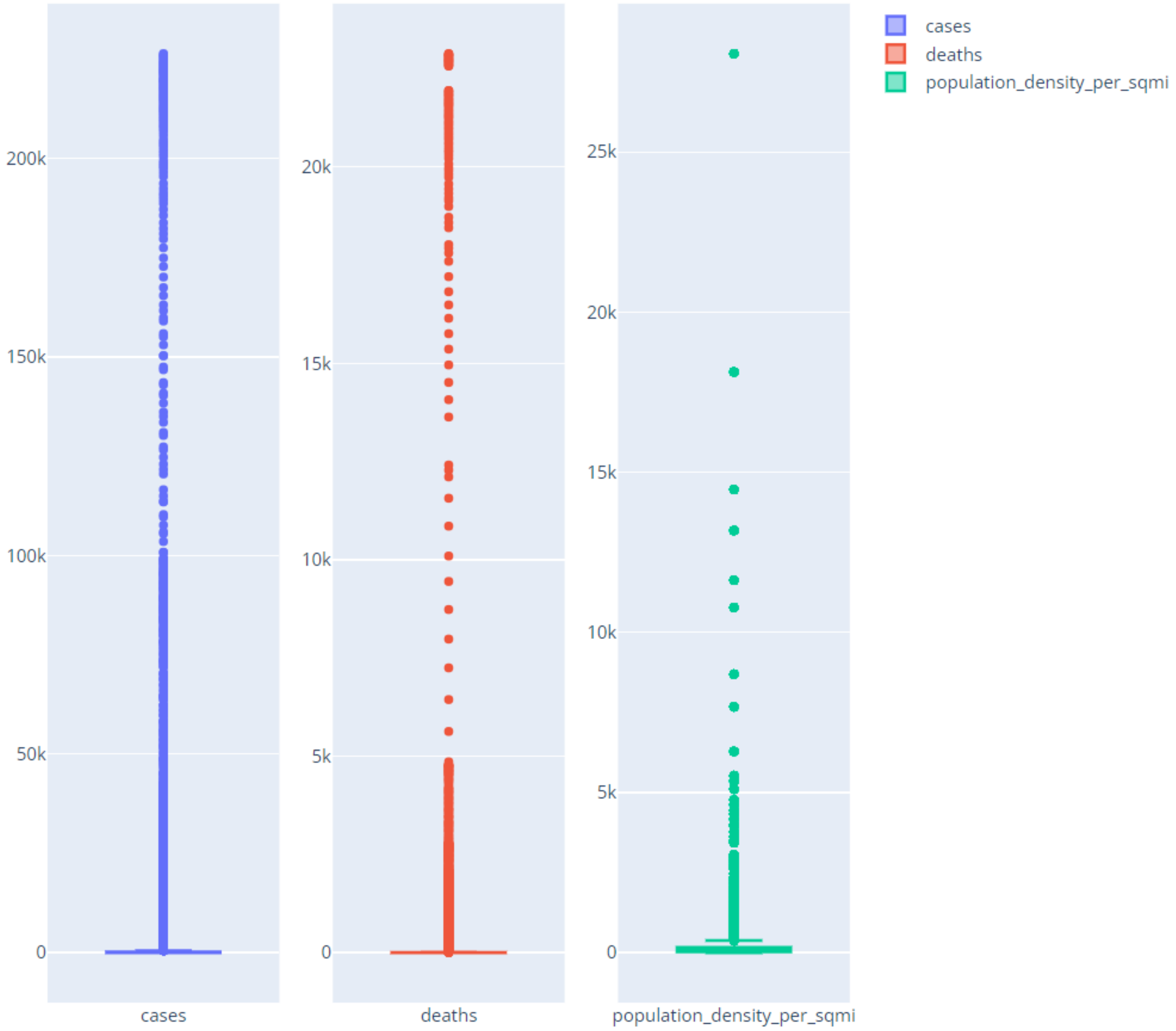


Figure 5. The 3 columns with the most outliers – impacting the dataset at large.

The above figure focuses on the three major columns and gives a graphical insight on the outliers in the situation.

Data Preparation

United States Covid-19 dataset has 790330 rows and 101 columns. We removed 22 columns that are highly correlated with other columns so we removed it.

Our data is not categorical to classify the target variable in the outcome. Our dataset has numeric value in 70 columns and the target variable is discrete numeric values we need to predict values from 0 to specific numbers in the dataset. The target variable is not a classification problem so we moved to the regression model. We applied 3 to 4 different models in our dataset. We split our dataset in 2 parts in the training set 80% and testing 20%. We assigned these models on standard scalar data and normal data in data frame. To apply regression, there is a need to convert categorical variables into numerical form. Here, we have changed the value of categorical variables to numerical form by using the revalue function.

Ultimately after multiple rounds of deliberation and reaching out to the owner of the dataset – we concluded down to 40 Attributes.

Data Partitioning

Since we have 650000 observations along with 40 attributes – it was best determined to go ahead with Stratified Cross Validation Method. In this particular method we needed to decide as valid number for the fold (generally between 5 to 10)

After consideration we decided to go ahead with 6 folds, each fold having 108333 observations. Here we went ahead with 4 folds being assigned for Train and validation and 2 folds for testing.

Thus, we ended up in near to 70-30 partition standard – with 67% of our data going for Train and Validation purpose and 33% of our data for testing purpose.

Data Mining Model/Methods

a. Linear Regression

Linear regression is a statistical method used to establish a relationship between two continuous variables, where one variable is known as the dependent variable, and the other is known as the independent variable. The goal of linear regression is to find the best-fitted straight line that represents the relationship between the two variables.

Advantages of Linear Regression

- Simple and easy to understand.
- Can be used to make predictions for new observations.
- Provides a measure of the strength and direction of the relationship between variables.
- Provides insight into the significance of variables in the relationship.
- Can be used for both simple and complex models.

Disadvantages of Linear Regression

- Assumes a linear relationship between variables, which may not always be the case.
- Sensitive to outliers, which can significantly affect the model's predictions.
- Cannot be used when the relationship between variables is non-linear.
- Assumes that the errors are normally distributed, with a constant variance across the range of the independent variable.

- Can be affected by multicollinearity, which occurs when two or more independent variables are highly correlated.

b. Ridge Regression

Ridge regression is a variant of linear regression that is used to handle multicollinearity, a condition in which the independent variables are highly correlated with each other. It introduces a penalty term to the cost function to shrink the regression coefficients, which can help reduce the impact of multicollinearity on the model's performance.

Advantages of Ridge Regression

- Helps reduce the impact of multicollinearity on the model's performance.
- Can handle a large number of predictors without overfitting the data.
- Can improve the stability and reliability of the model's predictions.
- Can handle non-normal or non-constant error terms.

Disadvantages of Ridge Regression

- The penalty term can reduce the interpretability of the regression coefficients.
- The choice of the penalty parameter can be subjective and may require tuning.
- Ridge regression assumes that all predictors are important, which may not always be the case.
- It cannot perform variable selection or feature elimination.

c. Lasso Regression

Lasso regression is another variant of linear regression that is used to handle multicollinearity, and it introduces a penalty term to the cost function to shrink the regression coefficients. However, unlike ridge regression, it uses an L1 penalty that can lead to some regression coefficients being set to zero, effectively performing variable selection.

Advantages of Lasso Regression

- Can perform variable selection or feature elimination, which can improve the interpretability of the model.
- Can handle a large number of predictors without overfitting the data.
- Can improve the stability and reliability of the model's predictions.
- Can handle non-normal or non-constant error terms.

Disadvantages of Lasso Regression

- The penalty term can reduce the interpretability of the regression coefficients.
- The choice of the penalty parameter can be subjective and may require tuning.
- Lasso regression may not perform well when the number of predictors is larger than the number of observations.
- Lasso regression assumes that the predictors are independent, which may not always be the case.

d. Gradient Boosting Regressor

Gradient Boosting Regressor (GBR) is a machine learning algorithm that uses a gradient boosting framework to fit a sequence of weak regression models to the data, with each model trying to correct the errors of the previous model. The final prediction is the sum of the predictions of all the models in the sequence.

Advantages of Gradient Boosting Regressor

- Can handle both numerical and categorical data.
- Can handle missing data without imputation.
- Can perform well on complex nonlinear relationships.
- Can handle high-dimensional data with feature selection.
- Can provide feature importance measures to aid interpretation.

Disadvantages of Gradient Boosting Regressor

- Can be computationally expensive and time-consuming.
- Can overfit the data if the model is too complex or the learning rate is too high.
- Can be sensitive to the choice of hyperparameters, such as the learning rate, number of trees, and maximum depth.
- Can be difficult to interpret due to the complexity of the model.

e. KNN Classifier

K-Nearest Neighbours (KNN) is a non-parametric machine learning algorithm used for classification and regression tasks. It works by identifying the k closest training examples in the feature space to a new data point and then assigning the class label of the majority of these neighbours to the new data point.

Advantages of K-Nearest Neighbours Classifier

- Simple and easy to understand.
- Does not require assumptions about the underlying data distribution.
- Can be used for both binary and multi-class classification tasks.
- Can be used with any distance metric.
- Can be updated easily with new data.

Disadvantages of K-Nearest Neighbours Classifier

- Can be sensitive to the choice of k, which may require tuning.
- Can be computationally expensive when the number of training examples is large.
- Can be affected by the curse of dimensionality, where the distance metric becomes less meaningful as the number of features increases.
- Cannot handle missing values or imbalanced classes well.
- Cannot handle irrelevant features, which can negatively affect the classification performance.

Data Model Results

As initially explained, we went with 4 folds for train and Validation. Each time the model runs on 3 different folds for training and One-fold for validation.
The results for each model are published below.

For reference,
*model***Train** : This gives the RMSE value of the Train folds
*model***T_R2** : This gives the R2 score of the train folds
*model***valid** : This gives the RMSE value of the Validation folds
*model***V_R2** : This gives the R2 score of the Validation folds

LinearRegressionTrain	**LinearRegressionT_R2**	**LinearRegressionvalid**	**LinearRegressionV_R2**
4288.841192	0.765374	4326.556394	0.763370
4294.266408	0.764979	4309.054821	0.764687
4298.732699	0.764947	4296.620364	0.764679
4307.496960	0.764601	4269.500173	0.765806

RidgeRegressionTrain	**RidgeRegressionT_R2**	**RidgeRegressionvalid**	**RidgeRegressionV_R2**
4288.841192	0.765374	4326.556391	0.763370
4294.266408	0.764979	4309.054808	0.764687
4298.732699	0.764947	4296.620356	0.764679
4307.496960	0.764601	4269.500194	0.765806

LassoRegressionTrain	**LassoRegressionT_R2**	**LassoRegressionvalid**	**LassoRegressionV_R2**
4288.843212	0.765374	4326.562033	0.763369
4294.268400	0.764979	4309.028414	0.764690
4148.456949	0.764947	4296.609528	0.764680
4307.498980	0.764601	4269.529241	0.765803

GBRegressorTrain	**GBRegressorT_R2**	**GBRegressorvalid**	**GBRegressorV_R2**
3051.257860	0.881245	3152.611129	0.874361
3077.305191	0.879311	3065.780703	0.880886
3081.438734	0.879221	3082.902328	0.878849
3057.951456	0.881364	3132.012330	0.873972

KNNTrain	**KNNT_R2**	**KNNvalid**	**KNNV_R2**
181.64226	0.999578848	265.738737	0.99910802
177.34788	0.999600772	274.302622	0.99903346
179.79416	0.999588894	278.695033	0.99900803
178.52801	0.999593668	274.036088	0.99904795

Table 1. The Above tables represent the value for the Training Folds

As clearly seen the KNN model returns results with least RMSE value and a great R2 Score. Clearly KNN is the model to move forward with.

KNN Model

As explained earlier – the data was divided into 6 folds. 4 for Training and 2 for Testing. We then set 39 attributes for input features (X) and the corresponding target variable as CASES (y).

Hyperparameter tuning was performed to get the most viable value of K for the model. In this particular case we went ahead with the k = 3 value as it was in line with few low RMSE scores and having 3 nearest neighbors would help us get a better result for our KNN model.

Later on, the performance of the model was calculated based on RMSE and R2 score.

RMSE for Testing Folds turned out to be 270 (avg).
R2 Score for Testing folds was 0.998909

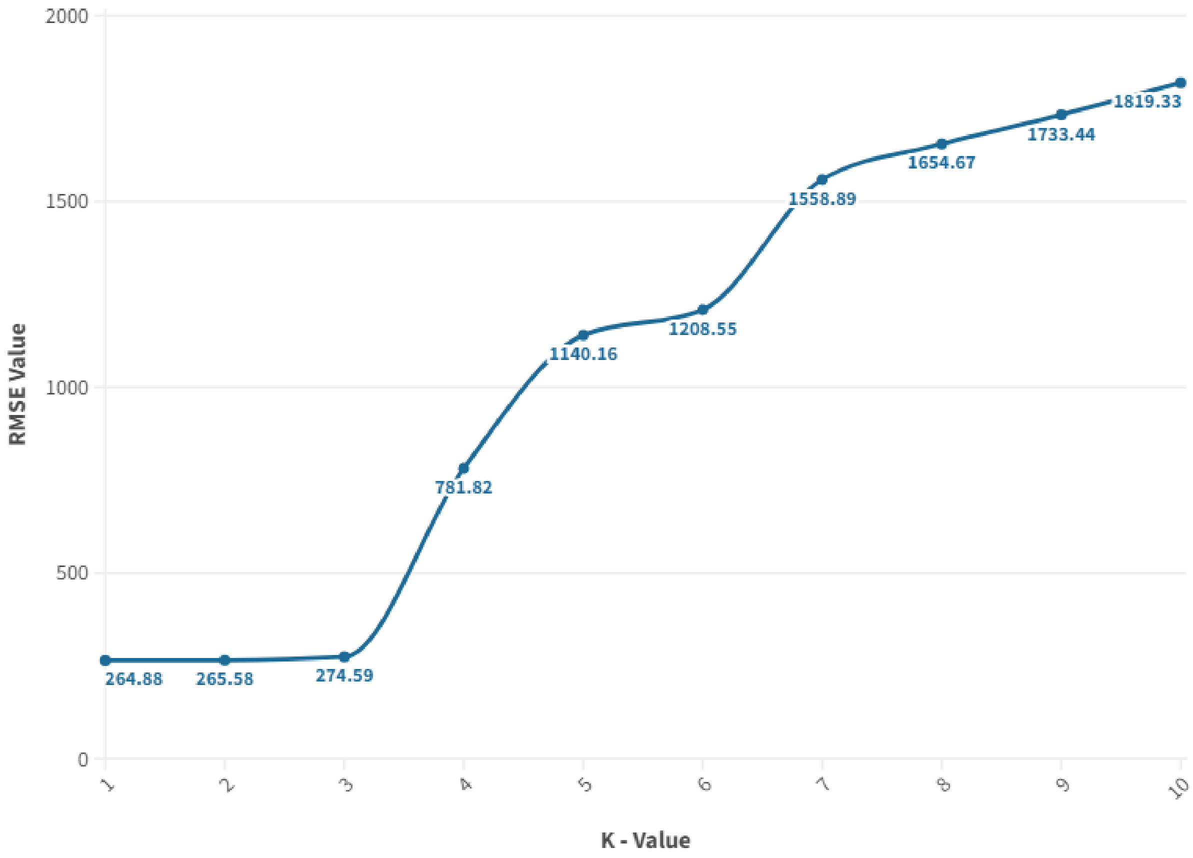


Figure 6. The Hyperparameter Tuning for finding the k value.

Choosing Non-Standardized data over Standardized data

In our dataset, as mentioned earlier we have dataset pertaining to outcomes that are factual and cannot be substituted or deleted as a result.
Standardization can be sensitive to outliers. If there are outliers in the dataset, standardization may cause them to have a larger impact on the model and lead to a worse performance.

This obviously impacted our process to move forward with the data. Over the course of running the model on standardized data we observed that the RMSE value was more for Standardized data rather than non-standardized data.
This can be attributed to the outliers in the dataset.

As clearly seen, most of the max values have a high difference compared to the mean of the particular attribute.

We can also assume another fact that Standardization can sometimes result in a loss of information, especially if the variables have a meaningful scale or units that are relevant to the problem at hand. If the information lost during standardization is important for the model to make accurate predictions, then it may result in a worse RMSE score.

Keeping all the above information in mind it became crucial for us to move ahead with the original data rather than standardizing it.

Even then the model was still impacted with the outliers – which was pretty evident in case of linear regression, lasso regression, Ridge Regression Models.

KNN performed comparatively better with an RMSE score of 270 (AVG)

Results based on RMSE and R2 score

R2 SCORE

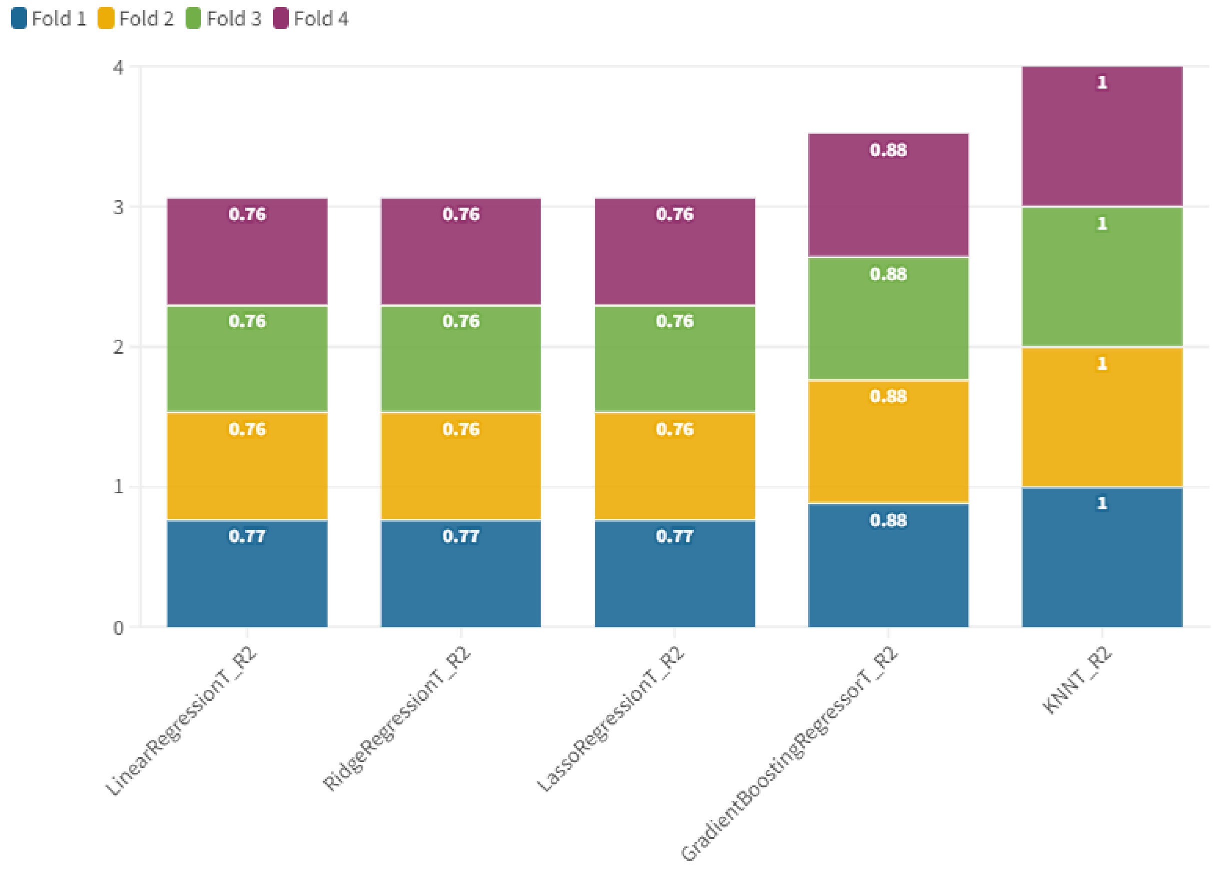


Figure 7. Visualization of the various Model Performance

As clearly visible KNN scored the highest R2 score followed by Gradient Boosting Regressor. Linear Regression, Ridge Regression and Lasso Regression came in close with their performances.

RMSE Score

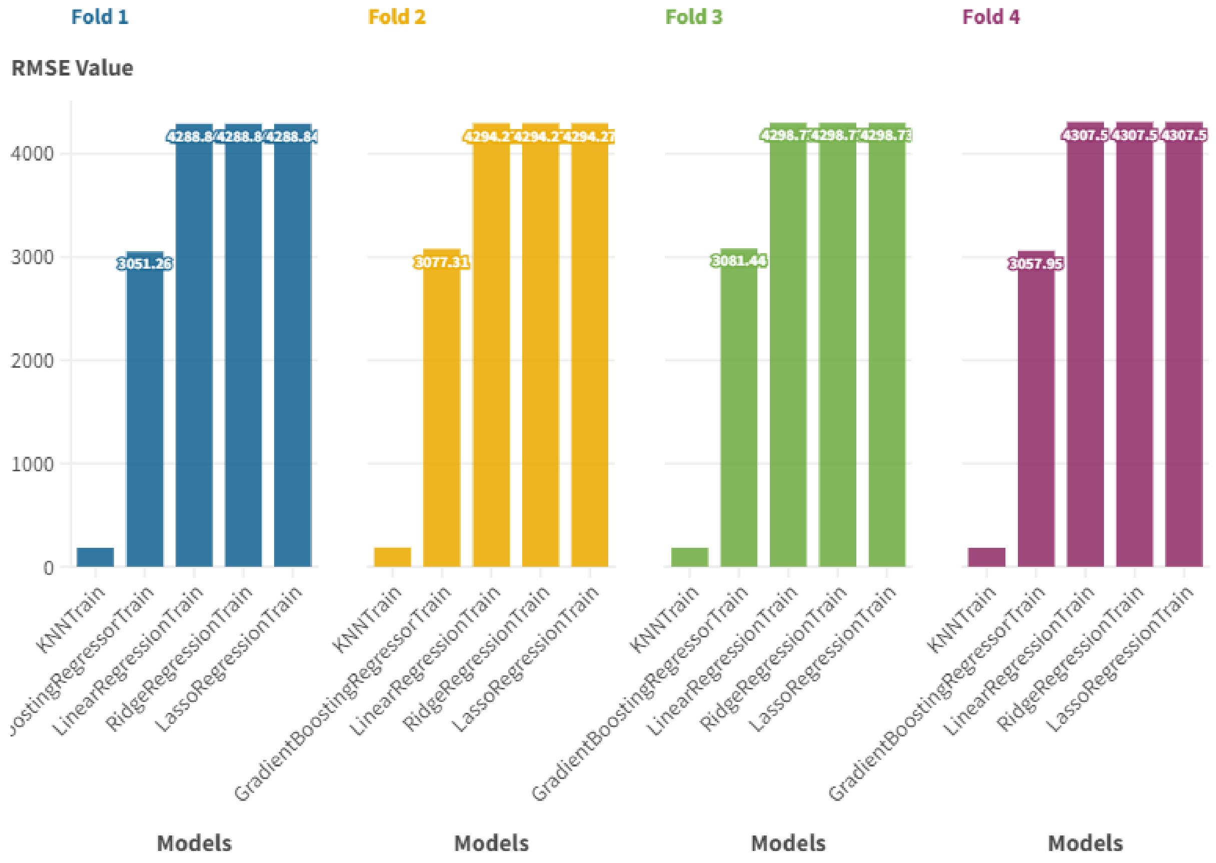


Figure 8. RMSE Scores

Clearly depictable through the chart, KNN had the least issue while predicting the target variable.
This in turn tells us that KNN was least affected by the outliers in the data whereas the regression models were heavily affected as they all scored an RMSE value of more than 4000.

FINAL vs PREDICTED VALUE

Our model running on KNN had an impressive R2 Score of 0.99 – this helped us in predicting the number of cases for most of the counties accurately.

Below is the graphical representation of the year 2020 – where we can see that the y predicted line stays in alignment to the number of actual cases for the most part of the trajectory.

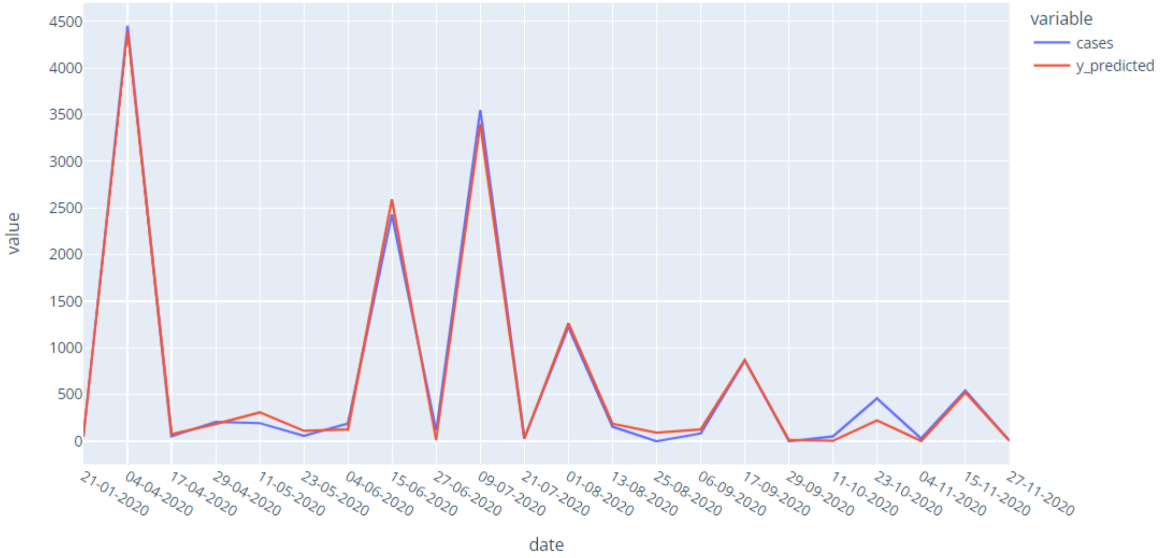
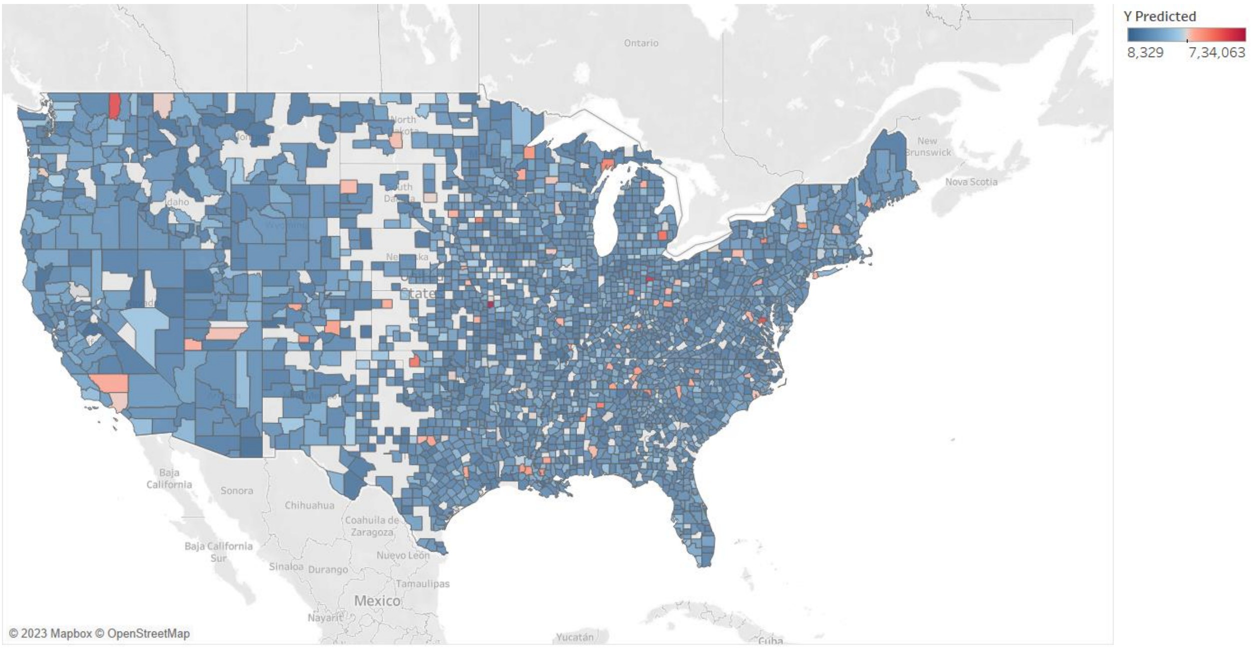


Figure 9. The Actual vs Predicated Cases Line Graph



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Y Predicted. Details are shown for State and County.

Figure 10. County wise Choropleth Map for Number of Cases Predicted.

IMPACT OF PROJECT OUTCOME

In the project we were able to successfully use various demographic attributes that impact a county and successfully train our model to predict the number of cases based on the 40 attributes that can drive the factors leading to the number of cases during an airborne disease pandemic.
We hope that this would help the government in identifying the resources necessary and the counties in need of the resources so as to prevent the spread of the disease and minimize the death rate in a county.

REFERENCES

[1] Dataset - https://www.kaggle.com/datasets/johnjdavisiv/us-counties-covid19-weather-sociohealth-data

[2] https://scikit-learn.org/stable/model_persistence.html - Pickle For saving Model Result

[3] https://machinelearningmastery.com/auto-sklearn-for-automated-machine-learning-in-python/ - Automation of Code to train data with Different models

[4] https://www.analyseup.com/python-machine-learning/stratified-kfold.html - Stratified Cross Validation